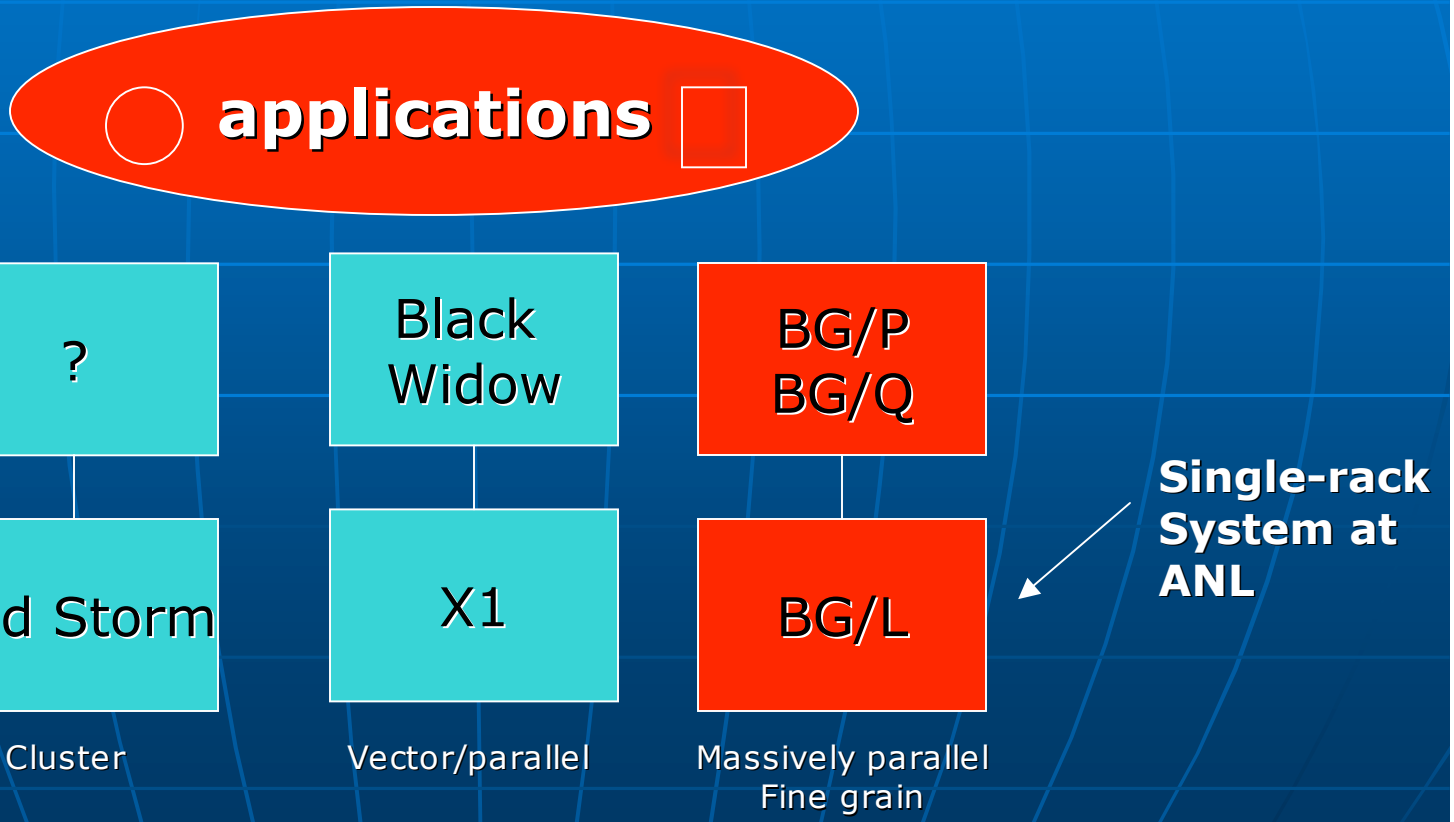
Initial Application Porting

Andrew Siegel
Katherine Riley
Argonne National Laboratory

Project Goals

- How will applications map onto different petaflop architectures?



Benchmarking BG/L

- Three layers of tests
  - Microbenchmarks
    - STREAM, mpptest, Euroben, Parkbench lmbench, SKaMPI, IO/ Tile test, HPC Challenge, Vector add and compiler options
  - Application kernel benchmarks
    - Petsc FUN3D, sPPM, UMT2000, NAS PB-MPI
- Web site constanly updated
  www-unix.mcs.anl.gov/~gropp/projects/parallel/BGL/index.htm

Benchmarking, cont.

- Application benchmarks
  - POP (Los Alamos Ocean Simulation)
  - QMC (monte carlo nucleonic forces)
  - Flash (Astrophyics -- hydro, burning, mhd, gravity)
  - Nek (Biological fluids – spectral element cfd)
  - Nimrod (Fusion – toroidal geometry)
  - pNeo (Nueroscience – Huxley nueron model)
  - Gyro (Plasma microturbulence)
  - IP
  - QCD (Lattice QCD)
  - Decartes, Ash, QGMG pending ..

Applications not Ported

- Require MIMD
  - Coupled ocean-atmosphere model
  - Coupled neutronics-hydro reactor model
- Codes with commercial components
  - e.g. Star-CD common for multiphase flow
- Codes with drivers written in Python

Application porting strategy

- Each application scientist gets 32-node dedicated partition for porting/tuning.
- Nightly full-rack reservations for bigger runs
- Mailing list with many contributors to help with porting, tuning, debugging issues.

Application expectations

- Current 1-rack system likely to do problems 1-2X size of our current Pentium/Myrinet Cluster
  - 1024 vs. 350 nodes
  - 2-3X performance / node on Pentium
  - Better scalability on BG/L
- Goal: scale to 10-20 rack system

Performance Measurements

Performance Matrix

app\metric	kernel	Communication pattern	Comp rate	Weak Scaling?	Threshold Memory	i/o	Primary Scalability bottlenecks
Flash	▪ Explicit hydro ▪ Multigrid ▪ SODE	dynamic ▪ Nearest Nghbr ▪ Global Ops	4/1	yes + αP^2	128Mb	O[1 Tb]	▪ Global block redistribution ▪ AMR multigrid
Nek5	Matrix-matrix product	fixed ▪ Nearest Nghbr ▪ 2 Global Ops	7/1	yes	16Mb	O[1 Tb]	none
QMC	Sparse matrix operations	fixed ▪ Master/slave	10/1	yes	O[Kb]	O[10 Mb]	non-scalable
pNeo	SODE	dynamic ▪ Neighborhood	1/2	yes+ non-trivial topology correction	1Gb	O[1 Mb]	Reduction operations over neighborhood
Columbus	Eigenvalue problem (Davidson method)	fixed ▪ Neighborhood	5/1	no	?	O[?]	?

Definitions
- *Comp Rate*: Estimated ratio of local work to communication time for RAM=256Mb
- *Weak Scaling:* Yes if ratio computation/comm constant at fixed local work
- *Threshold memory*: Local system memory where communication time = local work time

Application Performance

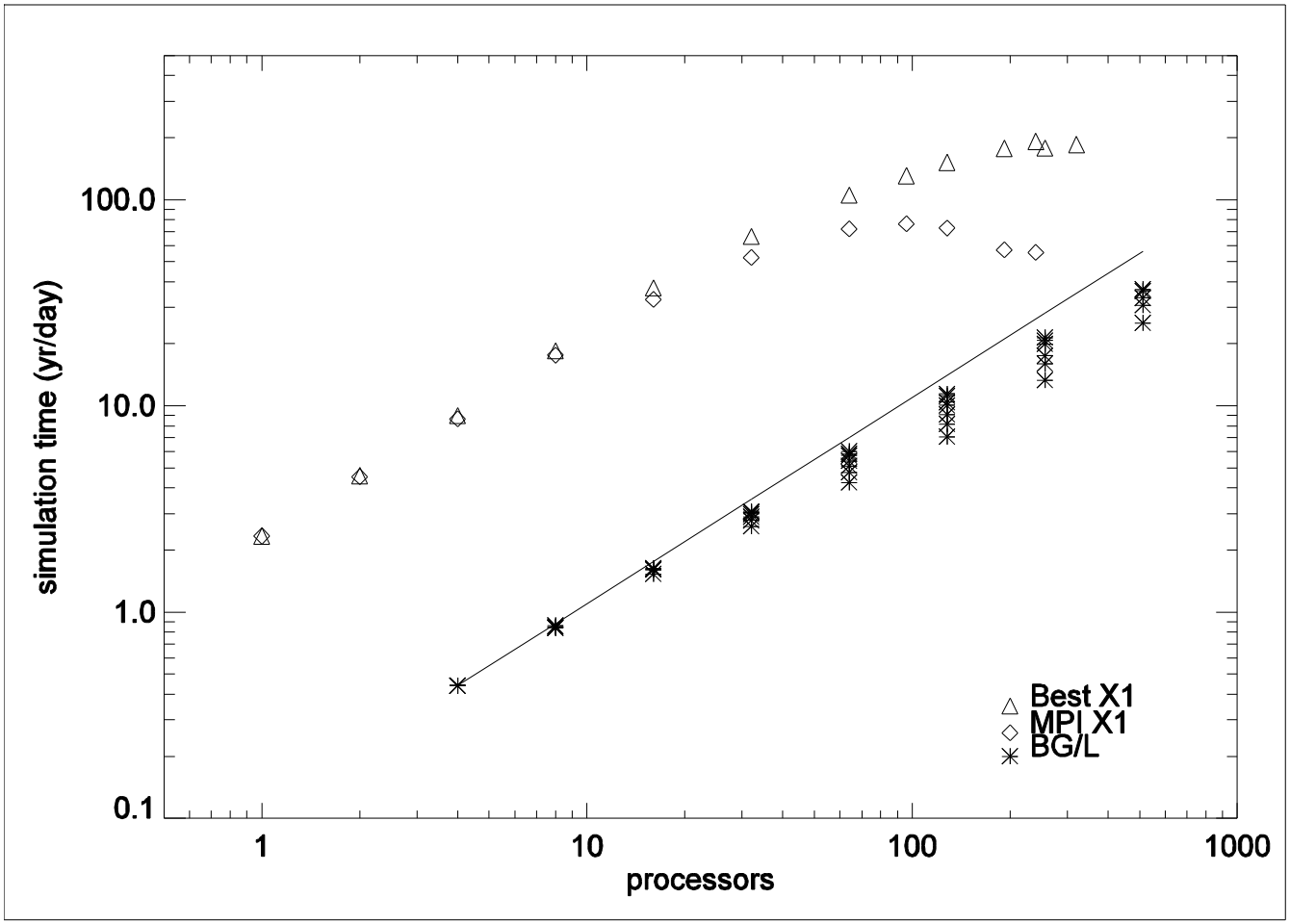
- General observations
  - Porting much easier than expected
    - Most programs have run extensively on NERSC mach
  - Single proc performance on poor end of expectations
    - No use of double-hummer
    - Uncertainty about data alignment issues
    - Loop unrolling limits give larger variations than we typically experience
    - One case of slow math intrinsics (using libm)
    - No essl
    - Addicted to hpmlib feedback to diagnose performance!
    - -qdebug=diagnostic doesn’t work on our system

Application Performance

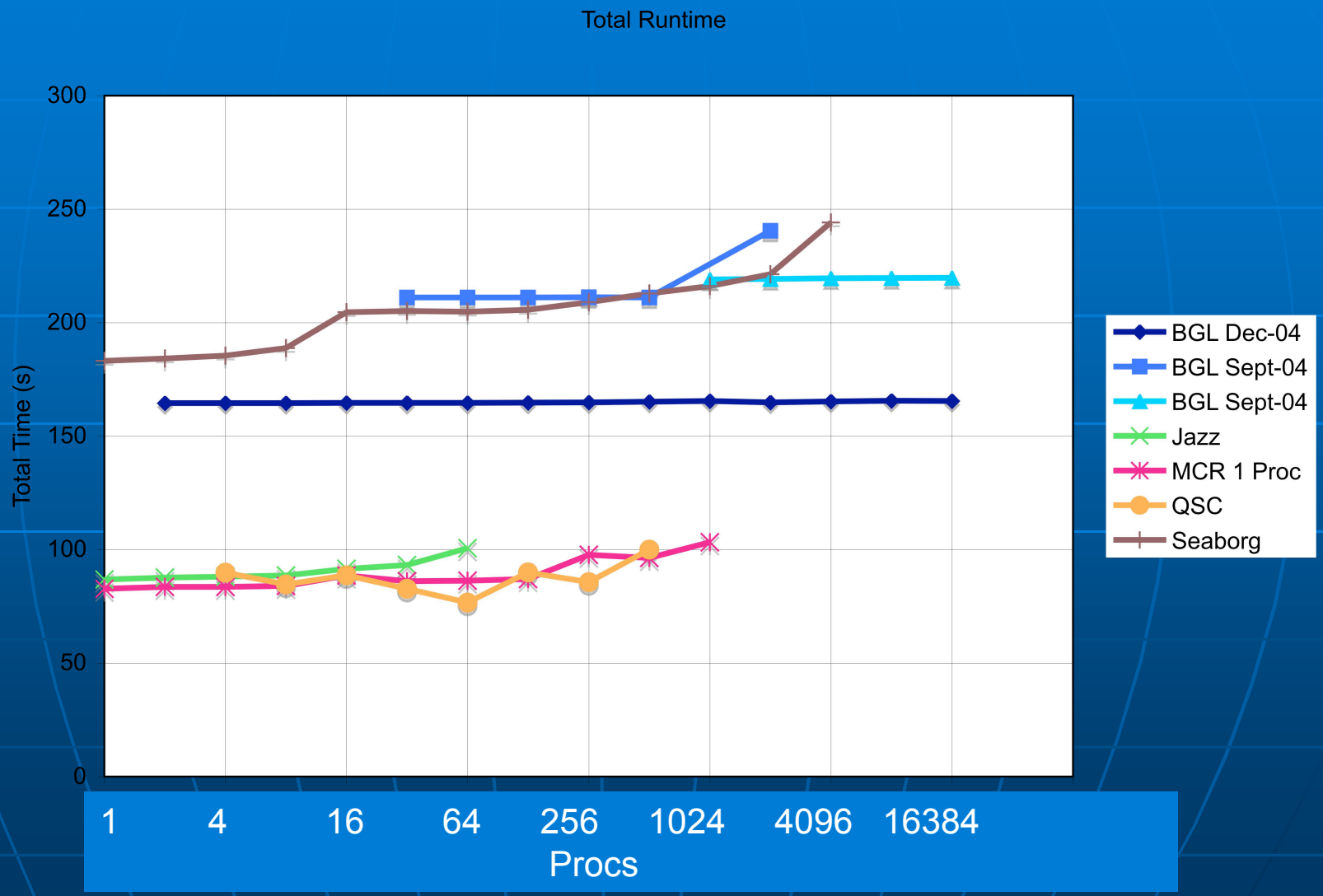
- General Observations, cont.
  - Network performance
    - Appears to be very good compared to what we're used to
    - Extremely reproducible timings
    - Still lots of detailed tests to run
  - VN mode
    - Most applications have at least one interesting problem which can be run with ½ the memory
  - IO
    - Haven't stressed it much at apps level

Some preliminary performance

POP Test


simulation time (yr/day)
100.0
10.0
1.0
0.1
1
10
100
1000
processors
Best X1
MPI X1
BG/L

Total Time For 2D Sod



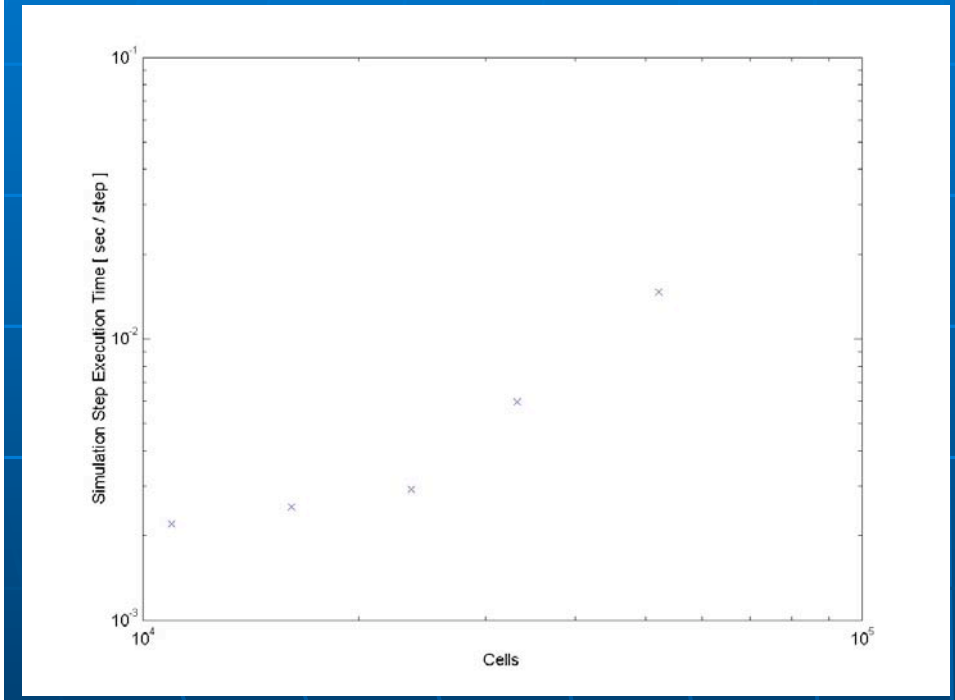
Ported tools/frameworks

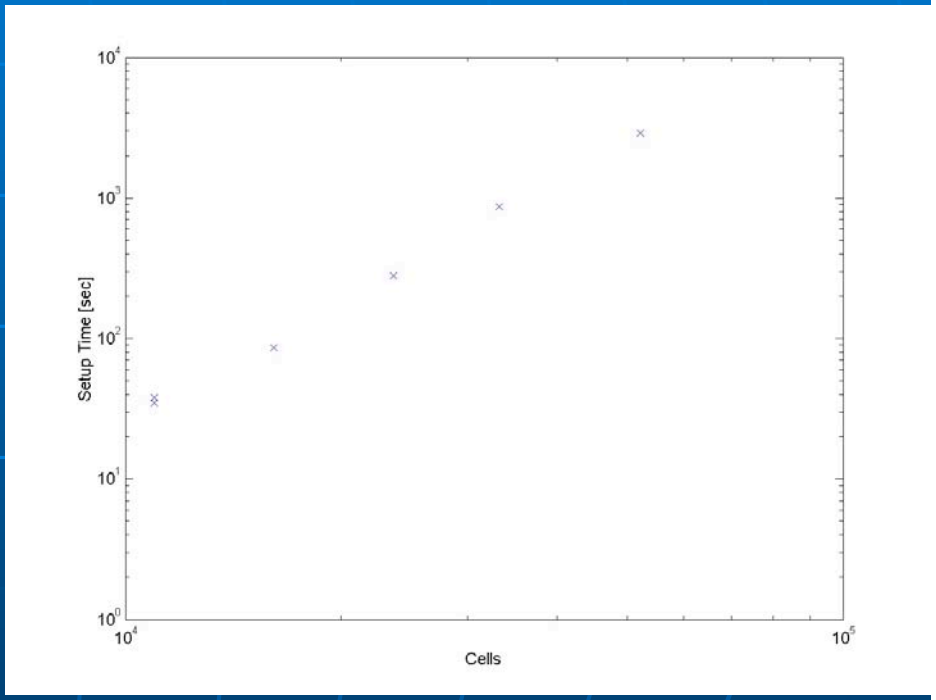
- TAU (U. or Oregon)
- PetsC
- fpmpi
- jumpshot

Summary of Application Needs

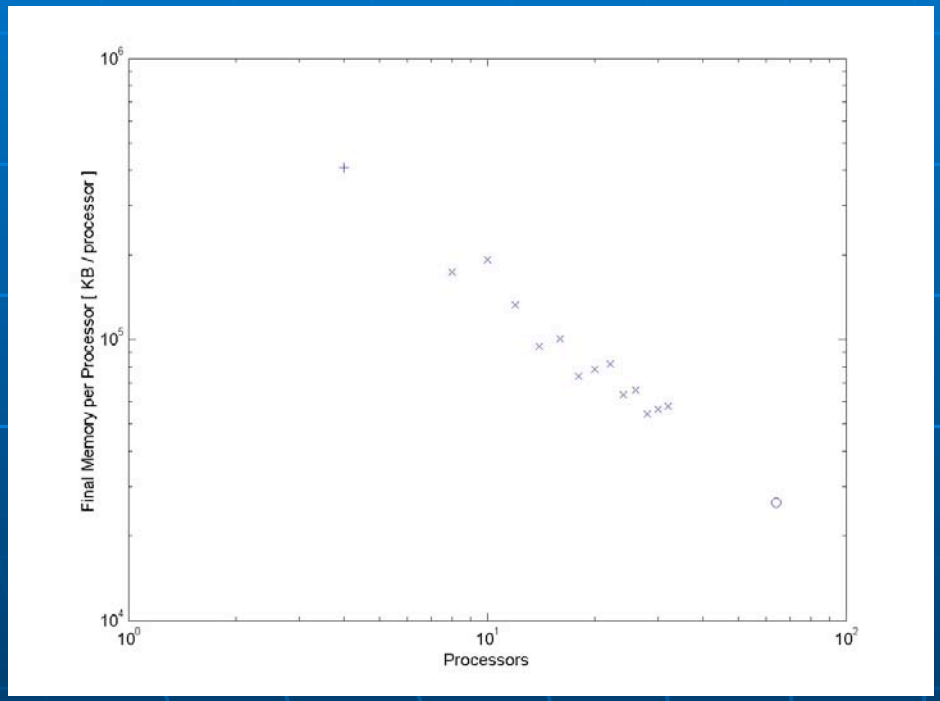
- Compilers
  - Double hummer assembly
  - Report functionality
  - Extended SIMD capabilities
  - Data alignment clarity
- Math Libraries
  - ESSL, mass(v), BLAS
- I/O : mpi i/o
  - hdf5, pNetcdf
- Debugger: gdb
- Profiler: gprof
- Software updates
  - Fixes to mpirun, compiler bugs
- HPM Lib | PAPI
- Stack/overwriting memory
- Better memory diagnostics (TAU?)
- General app requests
  - Dynamic libraries
  - MIMD possibilities
- Double FPU instructions

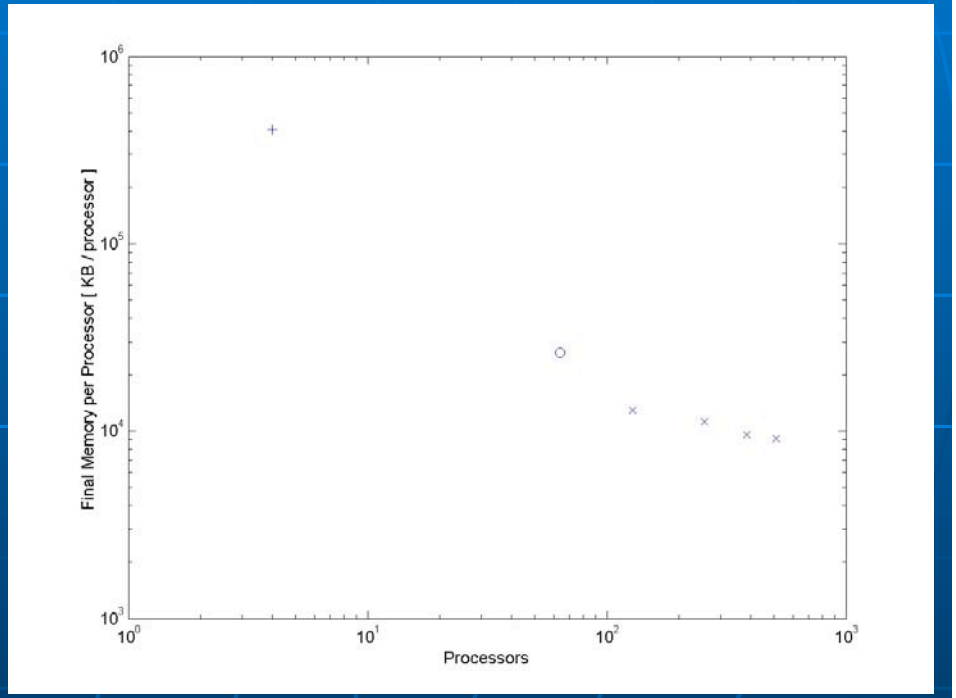
pNeo tests – problem size



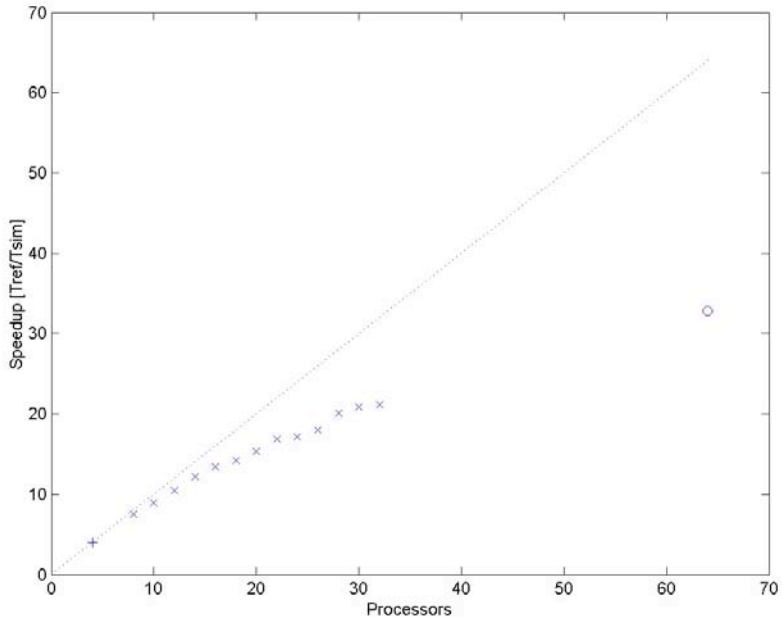


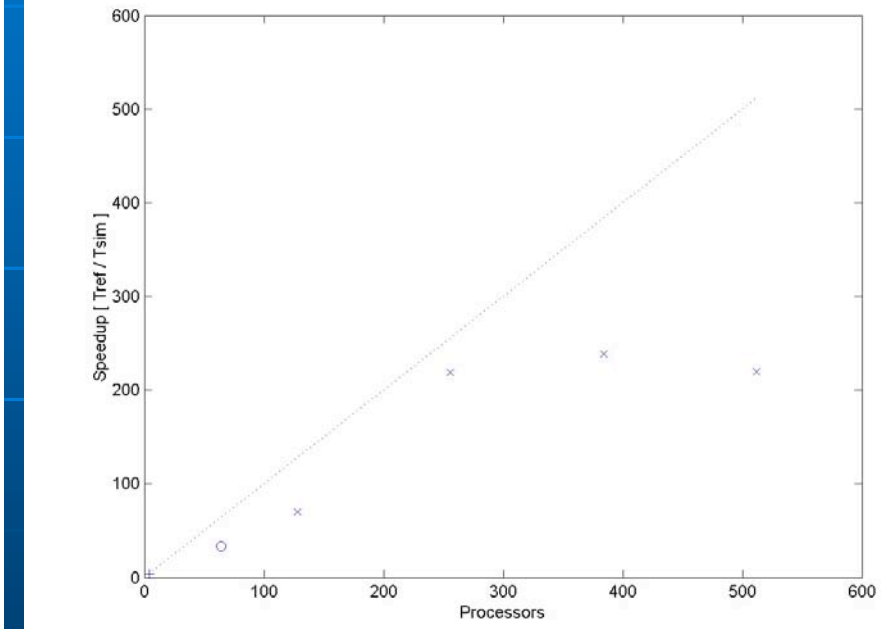
pNeo tests, cont.





pNeo tests, cont.





pNeo tests, cont.